

คู่มือเรียน เขียนโปรแกรม

python

(ภาคปฏิบัติ)

ฉบับอัปเดต
เพิ่มการเรียนรู้
โปรแกรม
ด้าน AI

- จุดเริ่มต้นสำหรับผู้สนใจการเขียนโปรแกรมด้วยภาษา Python ที่แม้แต่บริษัทยักษ์ใหญ่อย่าง Google ก็นำมาใช้งาน

- เรียนรู้ตั้งแต่คำสั่งพื้นฐานไปจนถึงการเขียนโปรแกรมเพื่อเชื่อมต่อกับระบบฐานข้อมูล

- สามารถนำไปใช้งานได้หลากหลาย แม้แต่ใช้เขียนโปรแกรมควบคุมหุ่นยนต์ก็ทำได้

- อธิบายโปรแกรมอย่างละเอียด เข้าใจทุกขั้นตอนในตัวอย่าง

```
def walk(x):
    return x+5

def dotwrite(last):
    nodename = getNodeName()
    label=symbol.symbolname.get(int(net[0]),net[0])
    print "%s (%label)" % (nodename, label)
    if isinstance(net[1], str):
        if net[1].strip():
            print "%s" % net[1]
        else:
            print ""
    else:
        print ""
        children = []
        for n, child in enumerate(last):
            children.append(dotwrite(child))
    return last

def write(net):
    print "%s (%label)" % (nodename, label)
    for name in children:
        print "%s" % name

def main():
    nodename = getNodeName()
    label=symbol.symbolname.get(int(net[0]),net[0])
    print "%s (%label)" % (nodename, label)
    if isinstance(net[1], str):
        if net[1].strip():
            print "%s" % net[1]
        else:
            print ""
    else:
        print ""
        children = []
        for n, child in enumerate(last):
            children.append(dotwrite(child))
    return last
```

สารบัญ

| | | |
|----------------|--|----|
| บทที่ 1 | ประโยชน์จากการใช้งาน Python | 9 |
| | ขั้นตอนการดาวน์โหลดและนำมาติดตั้งสำหรับ Microsoft Windows และ Linux | 11 |
| | ขั้นตอนการทดสอบหลังทำการติดตั้งสำเร็จ | 14 |
| บทที่ 2 | โครงสร้างและลักษณะการเขียนโปรแกรม | 15 |
| | คำสั่ง import | 19 |
| | คำสั่ง from | 20 |
| | รูปแบบการเขียนด้วยวิธีที่ 1 | 21 |
| | รูปแบบการเขียนด้วยวิธีที่ 2 | 21 |
| | รูปแบบการเขียนด้วยวิธีที่ 3 : แบบ Object Oriented | 22 |
| | รูปแบบการเขียนด้วยวิธีที่ 4 : แบบ Object Oriented พร้อมกลุ่มคำสั่งเรียกใช้งาน Object | 23 |
| | รูปแบบการเขียนด้วยวิธีที่ 5 : แบบ Object Oriented ที่เกิดจากรหัสที่ 3 และ 4 | 24 |
| | ภาพประยุกต์ใช้งานร่วมกัน | 24 |
| | การกำหนดค่าสำคัญย่อให้กับคำสั่งหลัก (Indenting) | 26 |
| | การนำความเข้าใจการใช้เครื่องหมาย " หรือ " สำหรับข้อความหรือดึงแปรชนิดข้อความ | 27 |
| | การใส่เครื่องหมาย (), [], { } ร่วมกับตัวแปรหรือฟังก์ชัน | 28 |
| | ประเภทการใช้เครื่องหมายร่วมกับตัวแปร | 28 |
| | ประเภทการใช้เครื่องหมายร่วมกับไพเรซีเจอร์หรือฟังก์ชัน | 29 |
| | ประเภทการใช้เครื่องหมายร่วมกับคำสั่งหรือฟังก์ชันภายใน | 30 |
| | ประเภทการใช้เครื่องหมายที่ทำหน้าที่เสมือนเป็นฟังก์ชันภายในแบบอัตโนมัติ | 30 |
| | การใส่เครื่องหมายคอมเม้นท์ (Comment) | 31 |
| | Python กับการใช้ตัวอักษรตัวเล็กหรือใหญ่ (Case Sensitive) | 31 |
| | นำทดลองเขียนโปรแกรมด้วยภาษา Python | 31 |
| | ทดลองเขียนคำสั่งด้วย Python Shell | 32 |
| | ทดลองเขียนคำสั่งด้วยรูปแบบไฟล์โปรแกรม | 37 |
| บทที่ 3 | รู้จักกับการใช้ตัวแปรของ Python | 39 |
| | การเลือกใช้งานตัวแปรสำหรับการเขียนโปรแกรม | 40 |
| | รูปแบบการสร้างและใช้งานตัวแปร | 40 |
| | ชนิดของตัวแปรที่มีใช้งานใน Python | 41 |
| | ทำความเข้าใจกับชนิดตัวแปรพื้นฐานของ Python | 42 |
| | ตัวแปรชนิดตัวเลขจำนวนเต็ม (Integer) | 43 |
| | ตัวแปรชนิดบูลีน (Boolean) | 45 |
| | ตัวแปรชนิดตัวเลขที่มีทศนิยม | 46 |
| | ตัวแปรชนิดตัวเลขจำนวนเชิงซ้อน (Complex Number) | 48 |
| | ตัวแปรชนิดตัวเลขทศนิยม Decimal | 48 |
| | ตัวแปรชนิดข้อความ (String) | 49 |

| | |
|---|-----|
| เทคนิคการ Slicing และ Striding String สำหรับตัวแปรชนิดข้อความ | 66 |
| การรีเซตค่าในตัวแปร | 73 |
| การใช้ตัวแปรประเภท Sequence | 73 |
| เทคนิคการใช้รวมตัวแปรประเภท Sequence ในลักษณะ Nested | 76 |
| รูปแบบการเขียนตัวแปรประเภท Sequence | 78 |
| ตัวแปรชนิด Tuple | 80 |
| ตัวแปรชนิด List | 81 |
| ตัวแปรชนิด Dictionary | 84 |
| การใช้โอเปอเรเตอร์ (Operator) กับตัวแปร | 89 |
| การใช้โอเปอเรเตอร์กับตัวแปรชนิดตัวเลข (Numeric And Iterable Operator) | 90 |
| การใช้โอเปอเรเตอร์กับตัวเลขจำนวนเต็มในระบิต (Integer Bitwise Operator) | 94 |
| เซตของข้อมูลกับการใช้โอเปอเรเตอร์ (Set Operator) | 96 |
| การกำหนดขอบเขตการใช้งานตัวแปรในโมดูล | 101 |
| บทที่ 4 การตรวจสอบเงื่อนไข | 109 |
| ผลลัพธ์เป็นไปตามเงื่อนไขหรือค่าที่เป็นจริง (True) | 110 |
| ผลลัพธ์ไม่ตรงกันเงื่อนไขที่ตรวจสอบค่าหรือได้ผลเป็นเท็จ (False) | 110 |
| การใช้คำสั่ง if ตรวจสอบเงื่อนไขและทำงานกลุ่มคำสั่งย่อย เมื่อการตรวจสอบเงื่อนไขเป็นจริง (True) เท่านั้น | 119 |
| การใช้คำสั่ง if ร่วมกับ elif สำหรับตรวจสอบเงื่อนไขและทำงานกลุ่มคำสั่งย่อยเมื่อผลลัพธ์การตรวจสอบเป็นจริง (True) เท่านั้น | 120 |
| การใช้คำสั่ง if ที่มีลักษณะการตรวจสอบเงื่อนไขแบบซ้อนกัน (Nested) | 123 |
| การใช้คำสั่ง if ร่วมกับคำสั่ง else | 127 |
| การกำหนดเงื่อนไขมากกว่าหนึ่งเงื่อนไขให้กับคำสั่ง if | 130 |
| รูปแบบการเขียนคำสั่ง if ร่วมกับ and operator | 130 |
| รูปแบบการเขียนคำสั่ง if ร่วมกับ or operator | 131 |
| รูปแบบการเขียนคำสั่ง if ร่วมกับ and และ or operator | 132 |
| บทที่ 5 คำสั่งการวนซ้ำ | 137 |
| ตัวอย่างที่ 1 คำสั่งรับข้อมูลแบบลูกทุ่งลูกทุ่ง | 137 |
| ตัวอย่างที่ 2 คำสั่งรับข้อมูลลูกไม้คำสั่งการวนซ้ำ for | 139 |
| ตัวอย่างที่ 3 การใช้คำสั่ง while สำหรับกลุ่มคำสั่งย่อยในการวนซ้ำซ้ำและสิ้นสุดด้วยการตรวจสอบเงื่อนไข | 140 |
| ตัวอย่างที่ 4 การใช้คำสั่ง while ที่มีมากกว่า 1 เงื่อนไขสำหรับการวนซ้ำซ้ำ | 141 |
| ตัวอย่างที่ 5 การใช้คำสั่ง while กับการกำหนดเงื่อนไขในลักษณะ Infinity Repetition | 142 |
| ตัวอย่างที่ 6 การใช้คำสั่ง for ร่วมกับ break และ continue | 147 |
| ตัวอย่างที่ 7 การใช้คำสั่ง while ร่วมกับ break และ continue | 148 |
| บทที่ 6 การสร้างโปรซีเจอร์หรือฟังก์ชันของ Python | 151 |
| รูปแบบการวางฟังก์ชันของ Python และวิธีเขียนโปรแกรมเพื่อเรียกใช้งานฟังก์ชันที่ใช้งานแบบ Global | 153 |
| | 155 |

| | |
|--|-----|
| ฟังก์ชันที่จัดการแบบ Local | 157 |
| Internal Method หรือ Private Method | 160 |
| Interface Method | 161 |
| การกำหนดขอบเขตของตัวแปรภายในข้อฟังก์ชัน | 164 |
| ลักษณะการใช้ชื่อวีรคดาร์กแวนต์และพารามิเตอร์ร่วมกับฟังก์ชัน (Argument and Parameter Function Keyword) | 168 |
| อาร์กิวเมนต์แบบตำแหน่งและจำนวนใช้งานที่แน่นอน (Positional Argument) | 169 |
| อาร์กิวเมนต์แบบขยายจำนวนตามค่าข้อมูล (Sequence Unpacking Operator) | 172 |
| ความแตกต่างระหว่างฟังก์ชันที่ไปและไม่ใช้คำสั่ง return | 175 |

| | |
|---|-----|
| บทที่ 7 การจัดการกับปัญหาเมื่อเกิดข้อผิดพลาดระหว่างการพัฒนาโปรแกรม (Error And Exceptions) | 179 |
| ข้อผิดพลาดที่เกิดจากการใช้คำสั่งผิดรูปแบบที่กำหนดไว้ (Syntax Error) | 179 |
| ข้อผิดพลาดที่เกิดระหว่างการทำงานของคำสั่ง (Runtime Error) | 180 |
| ข้อผิดพลาดที่เกิดจากการประมวลผลของกลุ่มคำสั่ง แต่ได้ผลลัพธ์ไม่ตรงตาม ความต้องการ (Logic Error) | 181 |
| คำสั่ง try และ except กับกรเกิด Error | 181 |
| การใช้คำสั่ง else ร่วมกับคำสั่ง try และ except | 192 |
| การใช้คำสั่ง finally ร่วมกับคำสั่ง try และ except | 195 |
| สร้างให้เกิด Error ด้วยคำสั่ง raise exception | 199 |

| | |
|---|-----|
| บทที่ 8 Python กับการเชื่อมต่อฐานข้อมูล | 201 |
| ประโยชน์ที่ได้จากการใช้ฐานข้อมูล | 202 |
| จะใช้ฐานข้อมูลด้วยวิธีอะไรบ้าง | 205 |
| แนวทางการต่อขอต่อข้อมูลในเชิงลึกกับฐานข้อมูล | 210 |
| มาเลือกใช้ฐานข้อมูลให้เหมาะกับงานที่กำลังพัฒนาโปรแกรม | 212 |
| กลุ่มฐานข้อมูลประเภท Flat File Database | 213 |
| การใช้ Database Manager (DBM) ร่วมกับไฟล์ข้อมูล | 219 |
| กลุ่มฐานข้อมูลประเภท File Server-base Database | 224 |
| การอ้างอิงโมดูลของ SQLite มาใช้งานในโมดูล Python | 226 |
| วิธีการสร้างไฟล์ฐานข้อมูลด้วยคำสั่ง DB-API ของ Python | 226 |
| การใช้ Cursor Object กับทำอ่านค่าข้อมูลด้วยคำสั่ง Select | 228 |
| การใช้ Cursor Object กับการเพิ่มข้อมูลใหม่ในตารางด้วยคำสั่ง Insert | 232 |
| การใช้ Cursor Object กับการแก้ไขข้อมูลที่มีอยู่แล้วในตารางด้วยคำสั่ง Update | 234 |
| การใช้ Cursor Object กับการลบเรคคอร์ดข้อมูลจากตารางด้วยคำสั่ง Delete | 237 |
| การสร้างตาราง (Table) และความสัมพันธ์ของตาราง (Relationship) | 238 |
| สร้างตารางด้วยคำสั่ง DDL (Data Definition Language) | 239 |
| สร้างตารางด้วยโปรแกรมเชิงมือสำหรับไฟล์ฐานข้อมูลประเภท SQLite | 246 |
| การเพิ่มเรคคอร์ดลงในตารางด้วยคำสั่ง insert ของ SQL Statement | 247 |
| การอ่านค่าข้อมูลจากตารางโดยใช้คำสั่ง Select | 254 |
| การแก้ไขค่าข้อมูลเดิมในตารางด้วยคำสั่ง Update | 259 |

| | |
|--|-----|
| การลบค่าข้อมูลในตารางข้อมูลด้วยคำสั่ง Delete | 262 |
| การปิดการเชื่อมต่อฐานข้อมูล | 267 |
| กลุ่มฐานข้อมูลประเภท Database Server | 267 |
| การดาวน์โหลด MySQL Server พร้อมชุด Workbench ของ MySQL | |
| และโมดูลพิเศษสำหรับเชื่อมต่อกับ Python | 269 |
| การติดตั้ง MySQL Server Database และชุดบริหารการจัดการฐานข้อมูล (MySQL Workbench) | 270 |
| การติดตั้งโมดูลพิเศษสำหรับการเชื่อมต่อฐานข้อมูลกับ Python | 275 |
| การสร้างฐานข้อมูลและตารางสำหรับการทดลองโปรแกรมด้วย MySQL Workbench | 275 |
| การพัฒนาโปรแกรมสำหรับเชื่อมต่อฐานข้อมูล MySQL ด้วย DB-API ของ Python | 293 |
| ปัญหาและวิธีการแก้ไขในระหว่างการพัฒนาโปรแกรม | 325 |
| บทที่ 9 พื้นฐานการใช้โมดูล Tkinter สำหรับ Graphical User Interface (GUI) | 328 |
| ทดสอบสร้างและปรับแต่งค่าใช้งานของรูปภาพประเภท GUI ด้วยโมดูล Tkinter | 334 |
| ตัวอย่างที่ 1 แสดงการสร้างรูปภาพจากโมดูล Tkinter | 334 |
| ตัวอย่างที่ 2 แสดงการสร้างรูปภาพและปรับแต่งค่า Attribute ที่ชื่อ title และ geometry | 334 |
| ตัวอย่างที่ 3 แสดงการสร้างรูปภาพและอ่านค่าความกว้างและสูงจาก method เพื่อปรับการแสดงผลให้ตรงกับหน้าจอที่ใช้งานในปัจจุบัน | 335 |
| ตัวอย่างที่ 4 แสดงการสร้างรูปภาพพร้อมการเรียกใช้ฟังก์ชันใช้งานด้วยการ bind | 336 |
| ทดสอบสร้างและปรับแต่งค่าใช้งานของกลุ่ม widget ภายใน GUI ของโมดูล Tkinter | 336 |
| ตัวอย่างที่ 1 แสดงการสร้าง Label widget บนจอภาพ | 340 |
| ตัวอย่างที่ 2 แสดงการสร้าง Label widget พร้อมการรับตัวเลือก Option | 341 |
| ตัวอย่างที่ 3 แสดงการสร้าง Label widget ที่ใช้ร่วมกับรูปภาพจากไฟล์เพื่อแสดงบนจอภาพ | 342 |
| บทที่ 10 ก้าวสู่โลก AI ด้วย Python | 357 |
| การติดตั้งไลบรารีเพิ่มเติม | 357 |
| การตรวจจับใบหน้าในภาพ | 359 |
| การตรวจจับดวงตาในภาพ | 361 |
| การรับรู้เสียงพูด | 363 |